Typed Graph Models for Semi-Supervised Learning of Name Ethnicity

Delip Rao
Dept. of Computer Science
Johns Hopkins University
delip@cs.jhu.edu

David Yarowsky
Dept. of Computer Science
Johns Hopkins University
yarowsky@cs.jhu.edu

Abstract

This paper presents an original approach to semi-supervised learning of personal name ethnicity from typed graphs of morphophonemic features and first/last-name co-occurrence statistics. We frame this as a general solution to an inference problem over typed graphs where the edges represent labeled relations between features that are parameterized by the edge types. We propose a framework for parameter estimation on different constructions of typed graphs for this problem using a gradient-free optimization method based on grid search. Results on both in-domain and out-of-domain data show significant gains over 30% accuracy improvement using the techniques presented in the paper.

1 Introduction

In the highly relational world of NLP, graphs are a natural way to represent relations and constraints among entities of interest. Even problems that are not obviously graph based can be effectively and productively encoded as a graph. Such an encoding will often be comprised of nodes, edges that represent the relation, and weights on the edges that could be a metric or a probability-based value, and type information for the nodes and edges. Typed graphs are a frequently-used formalism in natural language problems including dependency parsing (McDonald et al., 2005), entity disambiguation (Minkov and Cohen, 2007), and social networks to just mention a few.

In this paper, we consider the problem of identifying a personal attribute such as ethnicity from only an observed first-name/last-name pair. This has important consequences in targeted advertising and personalization in social networks, and in gathering intelligence for business and government research. We propose a parametrized typed graph framework for this problem and perform the hidden attribute inference using random walks on typed graphs. We also propose a novel application of a gradient-free optimization technique based on grid search for parameter estimation in typed graphs. Although, we describe this in the context of person-attribute learning, the techniques are general enough to be applied to various typed graph based problems.

2 Data for Person-Ethnicity Learning

Name ethnicity detection is a particularly challenging (and practical) problem in Nigeria given that it has more than 250 ethnicities[1] with minor variations. We constructed a dictionary of Nigerian names and their associated ethnicity by crawling baby name sites and other Nigerian diaspora websites (e.g. onlinenigeria.com) to compile a name dictionary of 1980 names with their ethnicity. We retained the top 4 ethnicities – Yoruba, Igbo, Efik Ibibio, and Benin Edo[2]. In addition we also crawled Facebook to identify Nigerians from different communities. There are more details to this dataset that

[1] https://www.cia.gov/library/publications/the-world-factbook/geos/ni.html

[2] Although the Hausa-Fulani is a populous community from the north of Nigeria, we did not include it as our dictionary had very few Hausa-Fulani names. Further, Hausa-Fulani names are predominantly Arabic or Arabic derivatives and stand out from the rest of the ethnic groups, making their detection easier.

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:shortpapers, pages 514–518,
Portland, Oregon, June 19-24, 2011. ©2011 Association for Computational Linguistics

will be made available with the data itself for future research.

3 Random Walks on Typed Graphs

Consider a graph $G = (V, E)$, with edge set E defined on the vertices in V. A typed graph is one where every vertex v in V has an associated type $t_v \in \mathcal{T}_V$. Analogously, we also use edge types $\mathcal{T}_E \subseteq \mathcal{T}_V \times \mathcal{T}_V$. Some examples of typed edges and vertices used in this paper are shown in Table 1. These will be elaborated further in Section 4.

Vertices	POSITIONAL_BIGRAM, BIGRAM, TRIGRAM, FIRST_NAME, LAST_NAME, …
Edges	POSITION (POSITIONAL_BIGRAM → BIGRAM), 32BACKOFF (TRIGRAM → BIGRAM), CONCURRENCE (FIRST_NAME → LAST_NAME), …

Table 1: Example types for vertices and edges in the graph for name morpho-phonemics

With every edge type $t_e \in \mathcal{T}_E$ we associate a real-valued parameter $\theta \in [0, 1]$. Thus our graph is parameterized by a set of parameters Θ with $|\Theta| = |\mathcal{T}_E|$. We will need to learn these parameters from the training data; more on this in Section 5. We relax the estimation problem by forcing the graph to be undirected. This effectively reduces the number of parameters by half.

We now have a weighted graph with a weight matrix $\mathbf{W}(\Theta)$. The probability transition matrix $\mathbf{P}(\Theta)$ for the random walk is derived by noting $\mathbf{P}(\Theta) = \mathbf{D}(\Theta)^{-1}\mathbf{W}(\Theta)$ where $\mathbf{D}(\Theta)$ is the diagonal weighted-degree matrix, i.e, $d_{ii}(\Theta) = \sum_j w_{ij}(\Theta)$.

From this point on, we rely on standard label-propagation based semi-supervised classification techniques (Zhu et al., 2003; Baluja et al., 2008; Talukdar et al., 2008) that work by spreading probability mass across the edges in the graph. While traditional label propagation methods proceed by constructing graphs using some kernel or arbitrary similarity measures, our method estimates the appropriate weight matrix from training data using grid search.

4 Graph construction

Our graphs have two kinds of nodes – nodes we want to classify – called target nodes and feature nodes which correspond to different feature types. Some of the target nodes can optionally have label information, these are called seed nodes and are excluded from evaluation. Every feature instance has its own node and an edge exists between a target node and a feature node if the target node instantiates the feature. Features are not independent. For example the trigram `aba` also indicates the presence of the bigrams `ab` and `ba`. We encode this relationship between features by adding typed edges. For instance, in the previous case, a typed edge (32BACKOFF) is added between the trigram `aba` and the bigram `ab` representing the backoff relation. In the absence of these edges between features, our graph would have been bipartite. We experimented with three kinds of graphs for this task:

First name/Last name (FN_LN) graph

As a first attempt, we only considered first and last names as features generated by a name. The name we wish to classify is treated as a target node. There are two typed relations 1) between the first and last name, called CONCURRENCE, where the first and last names occur together and 2) Where an edge, SHARED_NAME, exists between two first (last) names if they share a last (first) name. Hence there are only two parameters to estimate here.

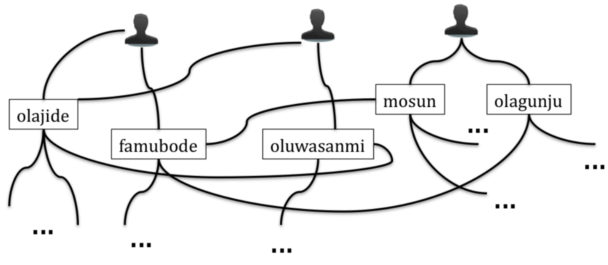

Figure 1: A part of the First name/Last name graph: Edges indicate co-occurrence or a shared name.

Character Ngram graph

The ethnicity of personal names are often indicated by morphophonemic features of the individual's given/first or family/last names. For example, the last names Polanski, Piotrowski, Soszynski, Sikorski with the suffix `ski` indicate Polish descent. Instead of writing suffix rules, we generate character n-gram features from names ranging from

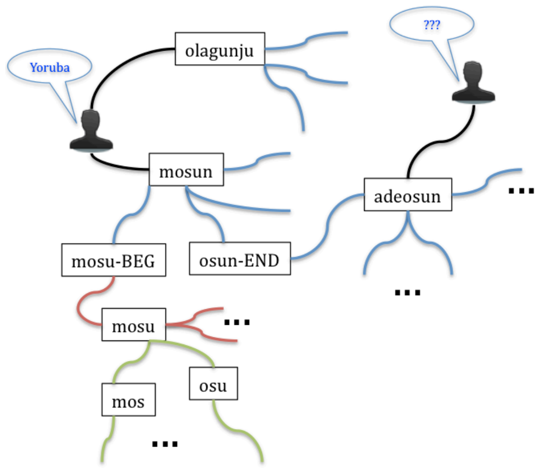

Figure 2: A part of the character n-gram graph: Observe how the suffix osun contributes to the inference of adeosun as a Yoruba name even though it was never seen in training. The different colors on the edges represent edge types whose weights are estimated from the data.

bigrams to 5-grams and all orders in-between. We further distinguish n-grams that appear in the beginning (corresponding to prefixes), middle, and end (corresponding to suffixes). Thus the last name, mosun in the graph is connected to the following positional trigrams mos-BEG, osu-MID, sun-END besides positional n-grams of other orders. The positional trigram mos-BEG connected to the position-independent trigram mos using the typed edge POSITION. Further, the trigram mos is connected to the bigrams mo and os using a 32BACKOFF edge. The resulting graph has four typed relations – 32BACKOFF, 43BACKOFF, 45BACKOFF, and POSITION – and four corresponding parameters to be estimated.

Combined graph

Finally, we consider the union of the character n-gram graph and the FirstName-LastName graph. Table 2 lists some summary statistics for the various graphs.

	#Vertices	#Edges	Avg. degree
FN_LN	22.8K	137.2K	3.6
CHAR. NGRAM	282.6K	1.2M	8.7
COMBINED	282.6K	1.3M	9.2

Table 2: Graphs for person name ethnicity classification

5 Grid Search for Parameter Estimation

The typed graph we constructed in the previous section has as many parameters as the number of edge types, i.e, $|\Theta| = |\mathcal{T}_E|$. We further constrain the values taken by the parameters to be in the range $[0, 1]$. Note that there is no loss of representation in doing so, as arbitrary real-valued weights on edges can be normalized to the range $[0, 1]$. Our objective is to find a set of values for Θ that maximizes the classification accuracy. Towards that effect, we quantize the range $[0, 1]$ into k equally sized bins and convert this to a discrete-valued optimization problem. While this is an approximation, our experience finds that relative values of the various $\theta_i \in \Theta$ are more important than the absolute values for label propagation.

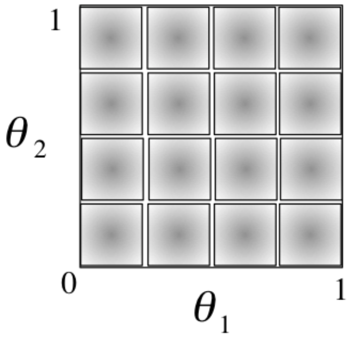

Figure 3: Grid search on a unit 2-simplex with $k = 4$.

The complexity of this search procedure is $O(k^n)$ for k bins and n parameters. For problems with small number of parameters, like ours ($n = 4$ or $n = 2$ depending on the graph model), and with fewer bins this search is still tractable although computationally expensive. We set $k = 4$; this results in 256 combinations to be searched at most and we evaluate each combination in parallel on a cluster. Clearly, this exhaustive search works only for problems with few parameters. However, grid search can still be used in problems with large number of edge types using one of the following two techniques: 1) Randomly sample with replacement from a Dirichlet distribution with same order as the number of bins. Evaluate using parameter values from each sample on the development set. Select the parameter values that result in highest accuracy on the development set from a large number of samples. 2) Perform a coarse grained search first using a small k on the range $[0, 1]$ and use that result to shrink the search range. Perform grid search again on this smaller range. We simply search exhaustively given the nature of our problem.

6 Experiments & Results

We evaluated our three different model variants under two settings: 1) When only a weak prior from the dictionary data is present; we call this 'out-of-domain' since we don't use any labels from Facebook and 2) when both the dictionary prior and some labels from the Facebook data is present; we call this 'in-domain'. The results are reported using 10-fold cross-validation. In addition to the proposed typed graph models, we show results from a smoothed-Naïve Bayes implementation and two standard baselines 1) where labels are assigned uniformly at random (UNIFORM) and 2) where labels are assigned according the empirical prior distribution (PRIOR). The baseline accuracies are shown in Table 3.

	Out-of-domain	In-domain
UNIFORM	25.0	25.0
PRIOR	42.6	42.6
Naïve Bayes	75.1	77.2

Table 3: Ethnicity-classification accuracy from baseline classifiers.

We performed similar in-domain and out-of-domain experiments for each of the graph models proposed in Section 4 and list the results in Table 4, *without* using grid search.

	Out-of-domain	In-domain
FN_LN	57.6	60.2
CHAR. NGRAM	73.2	76.8
%gain over FN_LN	27%	27.6%
COMBINED	77.1	78.7
%gain over CHAR. NGRAM	5.3%	2.5%

Table 4: Ethnicity-classification accuracy *without* grid search

Some points to note about the results reported in Table 4: 1) These results were obtained without using parameters from the grid search based optimization. 2) The character n-gram graph model performs better than the first-name/last-name graph model by itself, as expected due to the smoothing induced by the backoff edge types. 3) The combination of first-name/last-name graph and the n-gram improves accuracy by over 30%.

Table 5 reports results from using parameters estimated using grid search. The parameter estimation was done on a development set that was not used in the 10-fold cross-validation results reported in the table. Observe that the parameters estimated via grid search always improved performance of label propagation.

	Out-of-domain	In-domain
FN_LN	59.1	61.4
CHAR. NGRAM	76.7	78.5
COMBINED	**78.6**	**80.1**
Improvements by grid search (c.f., Table 4)		
FN_LN	2.6%	2%
CHAR. NGRAM	4.8%	2.2%
COMBINED	1.5%	1.7%

Table 5: Ethnicity-classification accuracy *with* grid search

7 Conclusions

We considered the problem of learning a person's ethnicity from his/her name as an inference problem over typed graphs, where the edges represent labeled relations between features that are parameterized by the edge types. We developed a framework for parameter estimation on different constructions of typed graphs for this problem using a gradient-free optimization method based on grid search. We also proposed alternatives to scale up grid search for large problem instances. Our results show a significant performance improvement over the baseline and this performance is further improved by parameter estimation resulting over 30% improvement in accuracy using the conjunction of techniques proposed for the task.

References

Shumeet Baluja, Rohan Seth, D. Sivakumar, Yushi Jing, Jay Yagnik, Shankar Kumar, Deepak Ravichandran, and Mohamed Aly. 2008. Video suggestion and discovery for youtube: taking random walks through the view graph. In *Proceeding of the 17th international conference on World Wide Web*.

Jonathan Chang, Itamar Rosenn, Lars Backstrom, and Cameron Marlow. 2010. epluribus: Ethnicity on social networks. In *Proceedings of the International Conference in Weblogs and Social Media (ICWSM)*.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Einat Minkov and William Cohen. 2007. Learning to rank typed graph walks: local and global approaches. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, New York, NY, USA. ACM.

Partha Pratim Talukdar, Joseph Reisinger, Marius Paşca, Deepak Ravichandran, Rahul Bhagat, and Fernando Pereira. 2008. Weakly-supervised acquisition of labeled class instances using graph random walks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the International Conference in Machine Learning*, pages 912–919.